CLAIMS

1. A software architecture for a distributed computing system comprising:

an application configured to handle requests submitted by remote devices over a network; and

an application program interface to present functions used by the application to access network and computing resources of the distributed computing system, the application program interface comprising various types related to constructing user interfaces.

2. A software architecture as recited in claim 1, wherein the various types comprise classes, interfaces, delegates, structures and enumerations.

3. A software architecture as recited in claim 1, wherein the distributed computing system comprises client devices and server devices that handle requests from the client devices, the remote devices comprising at least one client device.

4. A software architecture as recited in claim 1, wherein the distributed computing system comprises client devices and server devices that handle requests from the client devices, the remote devices comprising at least one server device that is configured as a Web server.

5. An application program interface embodied on one or more computer readable media, comprising: multiple types related to constructing user interfaces, the types comprising classes, interfaces, delegates, structures and enumerations.

6. An application program interface as recited in claim 5, wherein the classes comprise a forms class that represents a window or a dialog box that makes up an application's user interface.

7. An application program interface as recited in claim 6, wherein the forms class has multiple members comprising one or more of: public static properties, public static methods, public instance constructors, public instance methods, public instance properties, public instance events, protected instance properties, and protected instance methods.

8. An application program interface as recited in claim 5, wherein the type comprising the interfaces comprises a button control interface that allows a control to act like a button on a form.

9. An application program interface as recited in claim 5, wherein the type comprising the interfaces comprises a container control interface that provides functionality for a control to act as a parent for other controls.

10. An application program interface as recited in claim 5, wherein the type comprising the interfaces comprises an editing notification interface.

11. An application program interface as recited in claim 5, wherein the type comprising the interfaces comprises a data object interface that provides a format independent mechanism for transferring data.

12. An application program interface as recited in claim 5, wherein the type comprising the interfaces comprises a feature support interface that specifies a standard interface for retrieving feature information from a current system.

13. An application program interface as recited in claim 5, wherein the type comprising the interfaces comprises a message filter interface.

14. An application program interface as recited in claim 5, wherein the type comprising the interfaces comprises a handle-exposing interface to expose handles.

15. An application program interface as recited in claim 5, wherein the type comprising the interfaces comprises one or more of the following interfaces:

a button control interface that allows a control to act like a button on a form;

a container control interface that provides functionality for a control to act as a parent for other controls;

an editing notification interface;

a data object interface that provides a format independent mechanism for transferring data;

a feature support interface that specifies a standard interface for retrieving feature information from a current system;

a message filter interface; and

a handle-exposing interface to expose handles.

16. A distributed computer software architecture, comprising:

one or more applications configured to be executed on one or more computing devices, the applications handling requests submitted from remote computing devices;

a networking platform to support the one or more applications; and

an application programming interface to interface the one or more applications with the networking platform, the application programming interface comprising various types related to constructing user interfaces.

17. A distributed computer software architecture as recited in claim 16, , wherein the various types comprise classes, interfaces, delegates, structures and enumerations.

18. A distributed computer software architecture as recited in claim 17, wherein the classes comprises a forms class that represents a window or a dialog box that makes up an application's user interface.

19. A distributed computer software architecture as recited in claim 18, wherein the forms class has multiple members comprising one or more of: public static properties, public static methods, public instance constructors, public instance methods, public instance properties, public instance events, protected instance properties, and protected instance methods.

20. A distributed computer software architecture as recited in claim 17, wherein the type comprising the interfaces comprises a button control interface that allows a control to act like a button on a form.

21. A distributed computer software architecture as recited in claim 17, wherein the type comprising the interfaces comprises a container control interface that provides functionality for a control to act as a parent for other controls.

22. A distributed computer software architecture as recited in claim 17, wherein the type comprising the interfaces comprises an editing notification interface.

23. A distributed computer software architecture as recited in claim 17, wherein the type comprising the interfaces comprises a data object interface that provides a format independent mechanism for transferring data.

24. A distributed computer software architecture as recited in claim 17, wherein the type comprising the interfaces comprises a feature support interface that specifies a standard interface for retrieving feature information from a current system.

25. A distributed computer software architecture as recited in claim 17, wherein the type comprising the interfaces comprises a message filter interface.

26. A distributed computer software architecture as recited in claim 17, wherein the type comprising the interfaces comprises a handle-exposing interface to expose handles.

27. A distributed computer software architecture as recited in claim 17, wherein the type comprising the interfaces comprises one or more of the following interfaces:

a button control interface that allows a control to act like a button on a form;

a container control interface that provides functionality for a control to act as a parent for other controls;

an editing notification interface;

a data object interface that provides a format independent mechanism for transferring data;

a feature support interface that specifies a standard interface for retrieving feature information from a current system;

a message filter interface; and

a handle-exposing interface to expose handles.

28. A computer system including one or more microprocessors and one or more software programs, the one or more software programs utilizing an application program interface to request services from an operating system, the application program interface including separate commands to request services comprising services related to constructing user interfaces.

29. A method, comprising:

managing network and computing resources for a distributed computing system; and

exposing a set of functions that enable developers to access the network and computing resources of the distributed computing system, the set of functions comprising functions to facilitate construction of user interfaces

30. A method as recited in claim 29, further comprising receiving a request from a remote computing device, the request containing a call to the set of functions.

31. A method, comprising creating a namespace with functions that enable drawing and construction of user interfaces, the name space defining classes, interfaces, delegates, structures and enumerations.

32. A method as recited in claim 31, wherein the namespace defines a forms class that represents a window or a dialog box that makes up an application's user interface.

33. A method as recited in claim 32, wherein the forms class has multiple members comprising one or more of: public static properties, public static methods, public instance constructors, public instance methods, public instance properties, public instance events, protected instance properties, and protected instance methods.

34. A method as recited in claim 31, wherein the namespace defines an interface comprising a button control interface that allows a control to act like a button on a form.

35. A method as recited in claim 31, wherein the namespace defines an interface comprising a container control interface that provides functionality for a control to act as a parent for other controls.

36. A method as recited in claim 31, wherein the namespace defines an interface comprising an editing notification interface.

37. A method as recited in claim 31, wherein the namespace defines an interface comprising a data object interface that provides a format independent mechanism for transferring data.

38. A method as recited in claim 31, wherein the namespace defines an interface comprising a feature support interface that specifies a standard interface for retrieving feature information from a current system.

39. A method as recited in claim 31, wherein the namespace defines an interface comprising a message filter interface.

40. A method as recited in claim 31, wherein the namespace defines an interface comprising a handle-exposing interface to expose handles.